Active Learning and Search on Low-Rank Matrices

Dougal J. Sutherland

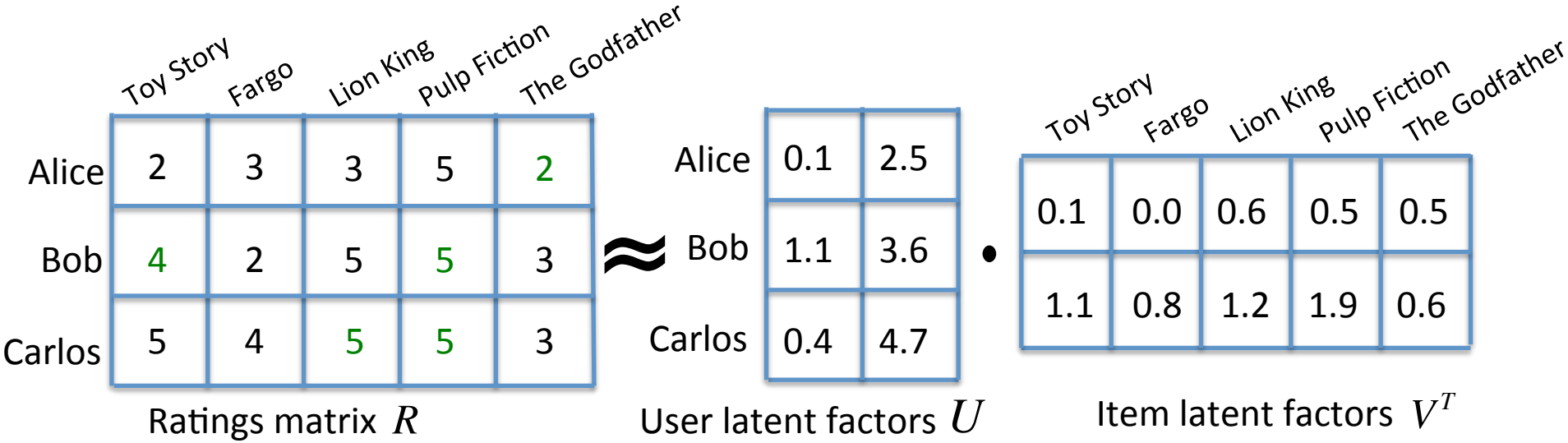
with Barnabás Póczos and Jeff Schneider

Carnegie Mellon University

Auton Lab

Collaborative prediction

- "Netflix problem": how can we predict whether users will like movies?
- Basic idea: similar users should have similar feelings about similar items
- Actually: assume the ratings matrix is low rank



Ratings matrix R User latent factors U Item latent factors V^T

Widely applicable

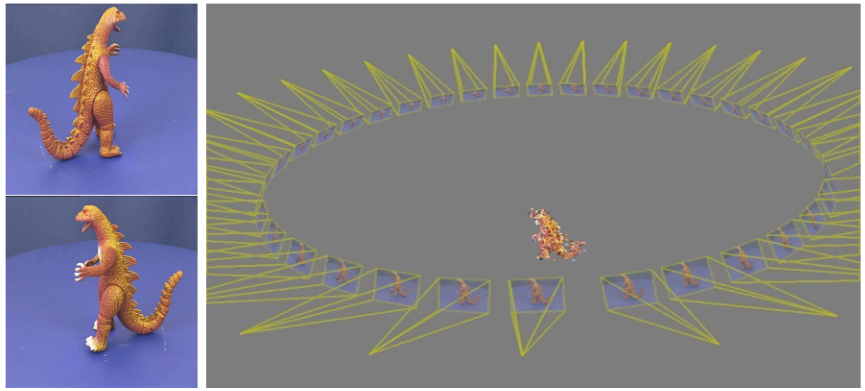
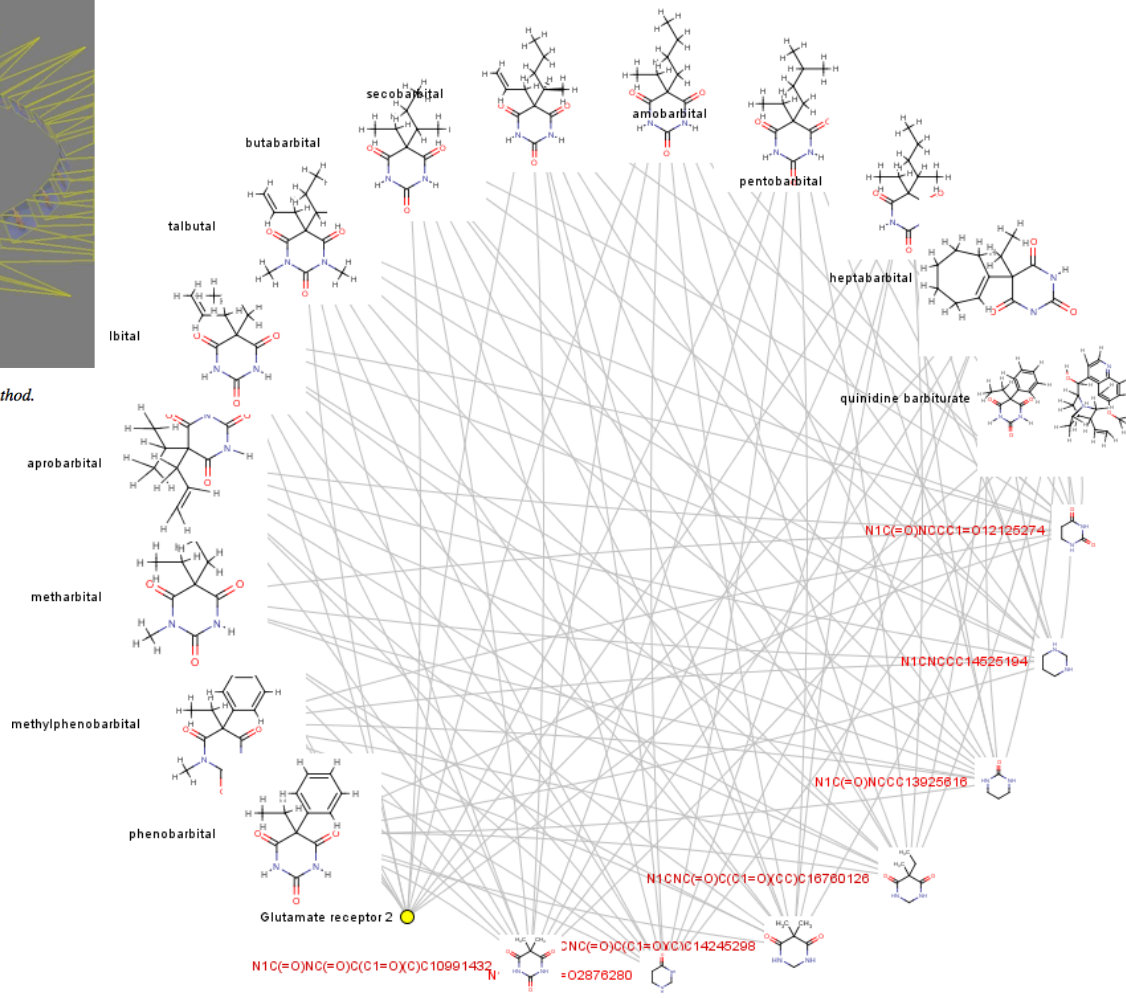


Figure 5. *Images from the dinosaur sequence, and the resulting reconstruction using our proposed method.*

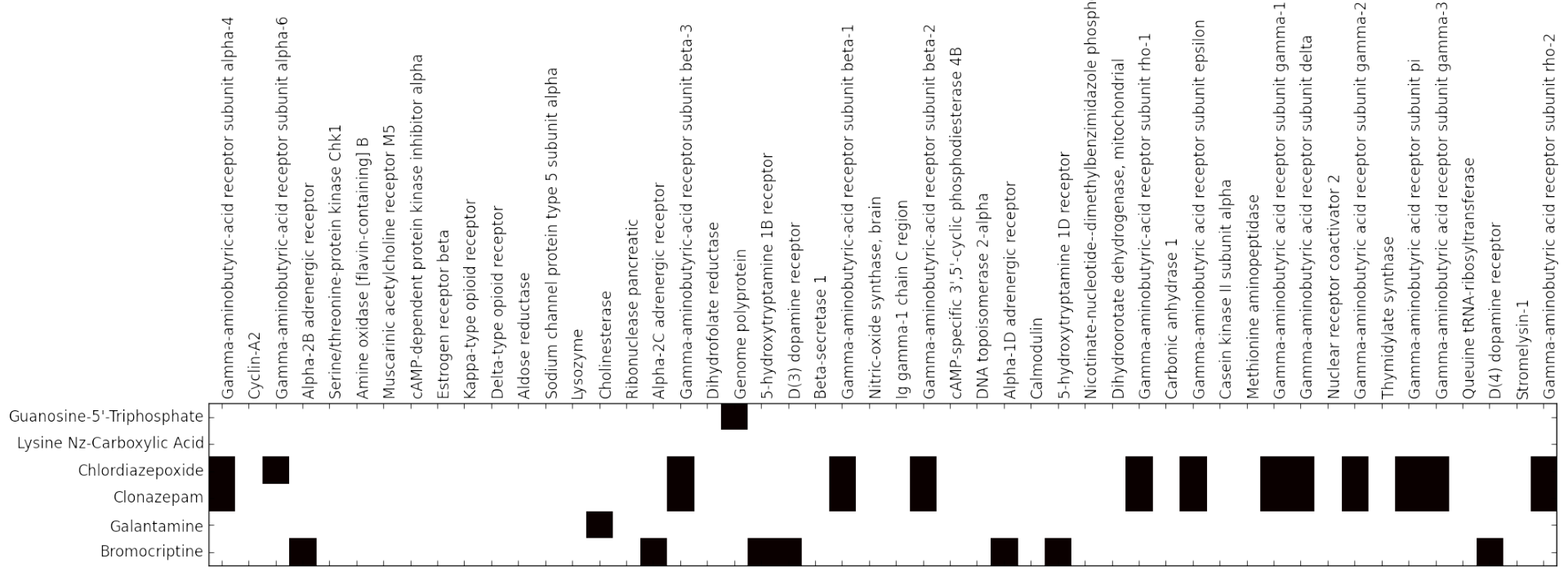
Erikkson & van den Hengel, CVPR 2010

Adams, Dahl, & Murray, UAI 2010

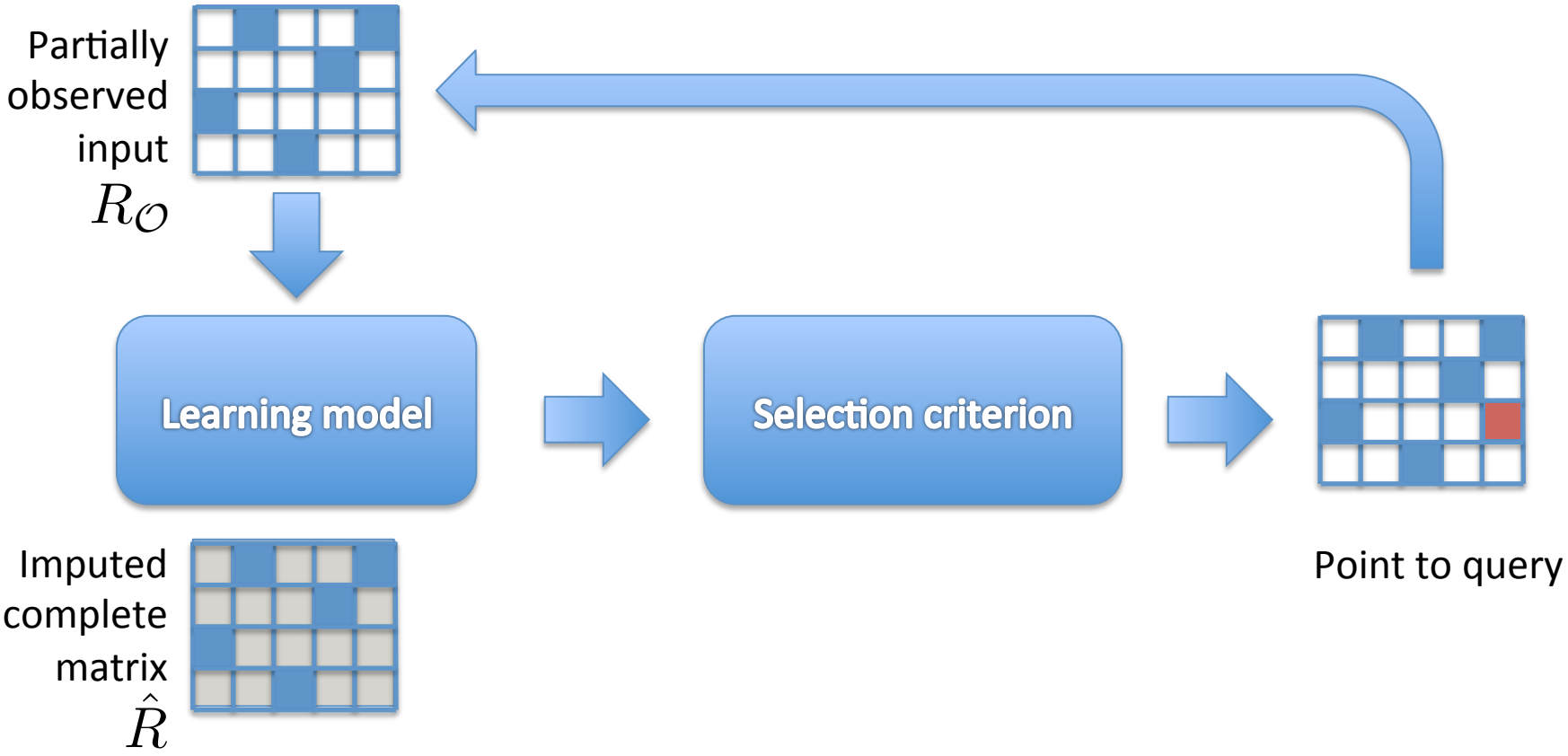


Active collaborative prediction

In practice, we rarely have a fixed training set. Sometimes we can choose to query specific points; we want the algorithm to tell us which ones to try.



Overall process



Learning goals

Prediction: minimize prediction error on unknown entries

$$\min \mathbb{E}\left[(R_{ij} - \hat{R}_{ij})^2 \mid (i,j) \notin \mathcal{O}\right]$$

Model: minimize uncertainty in the distribution of models

$$\min H\left[\text{model} \mid R_{\mathcal{O}}\right]$$

Magnitude Search: query largest-valued points possible

$$\max \sum_{(i,j)\in\mathcal{A}} R_{ij}$$

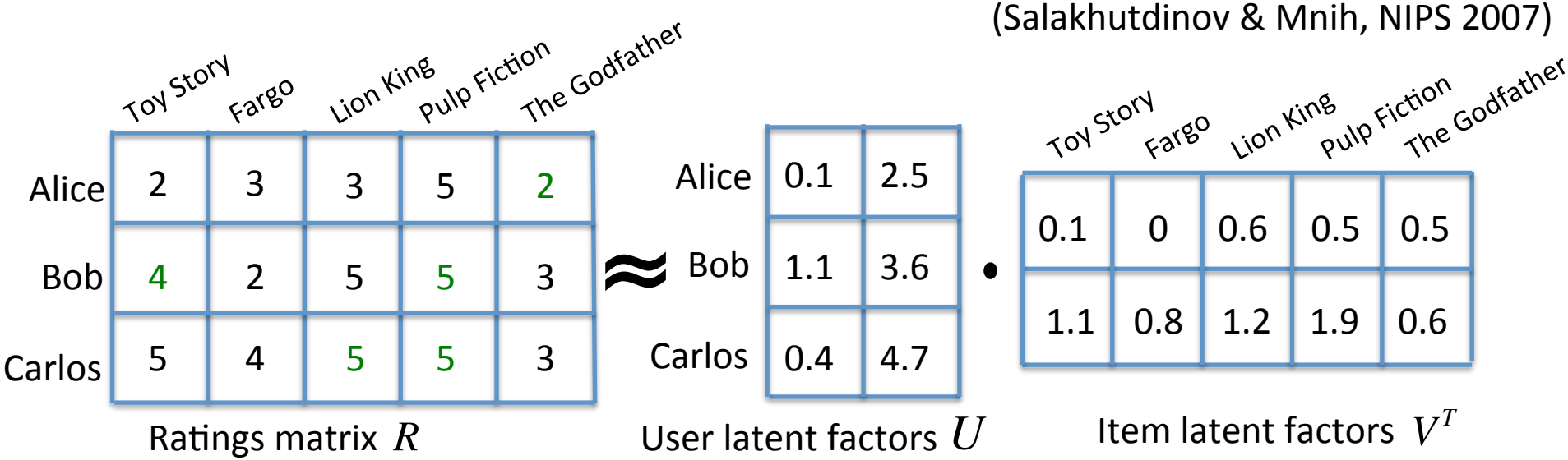
Search: query as many positive points as possible

$$\max \sum_{(i,j)\in\mathcal{A}} \mathbb{1}(R_{ij} \in +)$$

Probabilistic Matrix Factorization

Generative model for matrices of fixed rank *D*

(Salakhutdinov & Mnih, NIPS 2007)



$$R_{ij} \sim N\left(U_i^T V_j, \sigma^2\right) \qquad U_i \sim N\left(0, \sigma_U^2 I_D\right) \qquad V_j \sim N\left(0, \sigma_V^2 I_D\right)$$

$$-\ln p(U, V \mid R_{\mathcal{O}}) = \frac{1}{2\sigma^2}\|I \circ \left(R - UV^T\right)\|_F^2 + \frac{1}{2\sigma_U^2}\|U\|_F^2 + \frac{1}{2\sigma_V^2}\|V\|_F^2 + C$$

PMF Limitations

$$-\ln p(U, V \mid R_{\mathcal{O}}) = \frac{1}{2\sigma^2}\|I \circ (R - UV^T)\|_F^2 + \frac{1}{2\sigma_U^2}\|U\|_F^2 + \frac{1}{2\sigma_V^2}\|V\|_F^2 + C$$

- PMF is only really suited to a point estimate of *U*, *V*
- To do active learning, we need some information about our uncertainty in the model and/or the predictions

Variational PMF

One way to get posterior distribution info:

- Approximate joint distribution $p(U, V)$ with a parametric family $q(U, V)$
- Find best parameters by minimizing KL divergence

$$
\begin{aligned}
KL(q\|p) &= \int q(U,V) \ln \frac{q(U,V)}{p(U,V \mid R_{\mathcal{O}})} d\{U,V\} \\
&= -H[q] - \mathbb{E}_q \left[\ln p(U,V \mid R_{\mathcal{O}})\right] \\
&= -H[q] - C + \frac{1}{2\sigma_U^2} \sum_{i=1}^{N} \sum_{k=1}^{D} \mathbb{E}_q[U_{ik}^2] + \frac{1}{2\sigma_V^2} \sum_{j=1}^{M} \sum_{k=1}^{D} \mathbb{E}_q[V_{jk}^2] \\
&\quad + \frac{1}{2\sigma^2} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(\sum_{k=1}^{D} \sum_{\ell=1}^{D} \mathbb{E}_q[U_{ki} V_{kj} U_{\ell i} V_{\ell j}] - 2R_{ij} \sum_{k=1}^{D} \mathbb{E}_q[U_{ki} V_{kj}] + R_{ij}^2 \right)
\end{aligned}
$$

Variational PMF: full normal

- One option: normal over vector of entries in *U, V*
  - Expectations we need are in closed form (Isserlis' Thm.)
  - Can optimize with projected gradient descent
  - $O(D^2 (N+M)^2)$ memory, $O(D^3 (N+M)^3)$ time to project

Mean μ
$D(N+M)$

U_{11}	
U_{12}	
U_{21}	
U_{22}	
U_{31}	
U_{32}	
V_{11}	
V_{12}	
V_{21}	
V_{22}	

cov Σ
$(D(N+M))^2$

	U_{11}	U_{12}	U_{21}	U_{22}	U_{32}	U_{32}	V_{11}	V_{12}	V_{21}	V_{22}
U_{11}										
U_{12}										
U_{21}										
U_{22}										
U_{31}										
U_{32}										
V_{11}										
V_{12}										
V_{21}										
V_{22}										

Variational PMF: fully factorized

- Another: assume each element of U and V is independent
  - (Silva & Carin, KDD 2012)
  - O(D (N+M)) memory, projection is trivial

Mean μ
$D(N+M)$

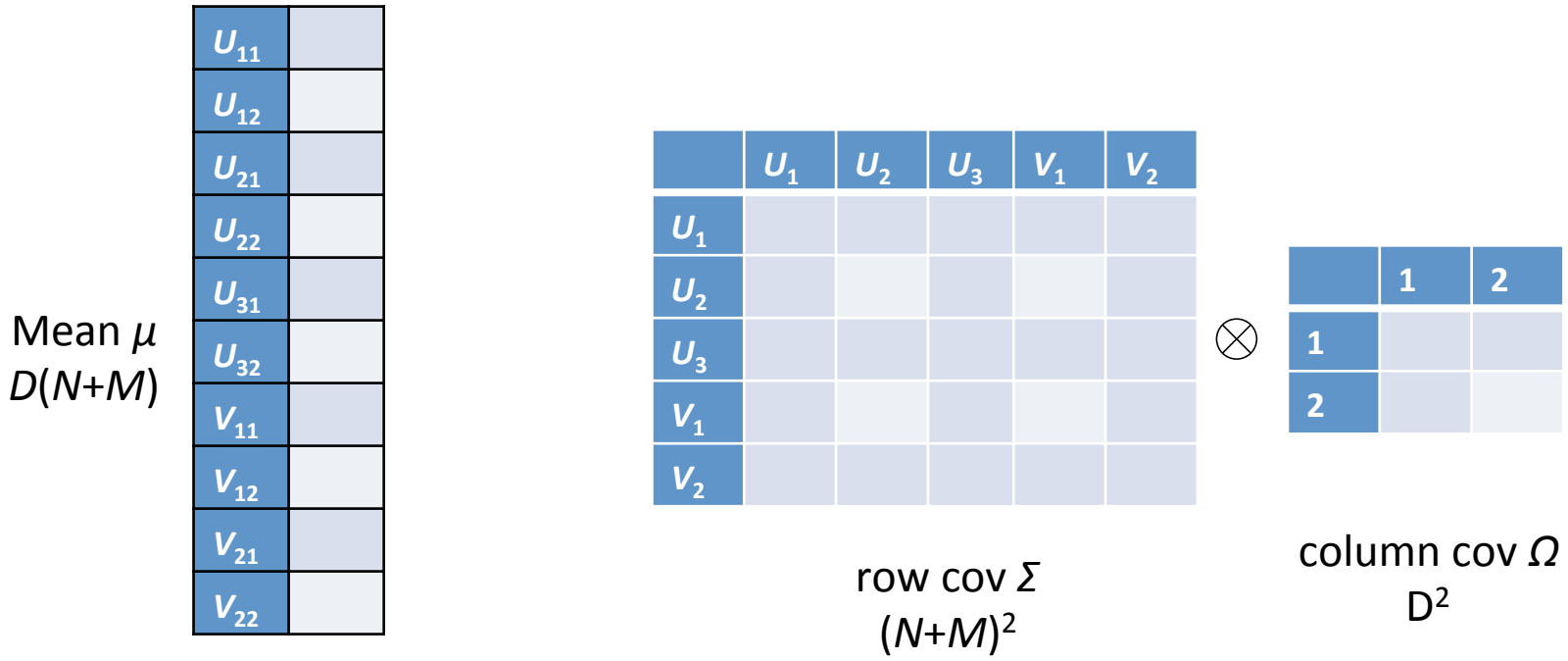
U_{11}
U_{12}
U_{21}
U_{22}
U_{31}
U_{32}
V_{11}
V_{12}
V_{21}
V_{22}

diagonal cov Σ
($D(N+M)$)

	U_{11}	U_{12}	U_{21}	U_{22}	U_{32}	U_{32}	V_{11}	V_{12}	V_{21}	V_{22}
U_{11}										
U_{12}										
U_{21}										
U_{22}										
U_{31}										
U_{32}										
V_{11}										
V_{12}										
V_{21}										
V_{22}										

Variational PMF: matrix normal

- In between: matrix normal over stacked *U, V*
  - Decompose cov into user/item covariance + latent d covariance
  - Expectations / gradient descent basically the same
  - $O(D^2 + (N+M)^2)$ memory, $O(D^3 + (N+M)^3)$ time to project



Markov chain Monte Carlo

Another way to get posterior info for PMF is to get samples from it (approximately, asymptotically…).

BPMF (Salakhutdinov & Mnih, ICML 2008) lets normal priors on U and V have arbitrary means/covariances, with Gaussian-Wishart hyperpriors.

- Can sample through Gibbs
- We use Hamiltonian MCMC with the **N**o-**U**-**T**urn **S**ampler (Hoffman & Gelman, JMLR in press)

Myopic selection criteria

- **Prediction:** element with highest variance (uncertainty sampling)

$$\arg\max_{(i,j)} \mathrm{Var}[R_{ij}]$$

- **Model:** ?

- **Magnitude search:** element with highest mean

$$\arg\max_{(i,j)} \mathbb{E}[R_{ij}]$$

- **Search:** element with highest probability of being positive

$$\arg\max_{(i,j)} \mathbb{P}[R_{ij} \in +]$$

Lookahead criteria

Integrate over possible outcomes (Garnett et al., ICML 2012)

$$\int_x \mathrm{d}\hat{\mathbb{P}}(R_{ij} = x)\, \mathbb{E}\left[f(q) \mid R_{\mathcal{O}}, R_{ij} = x\right]$$

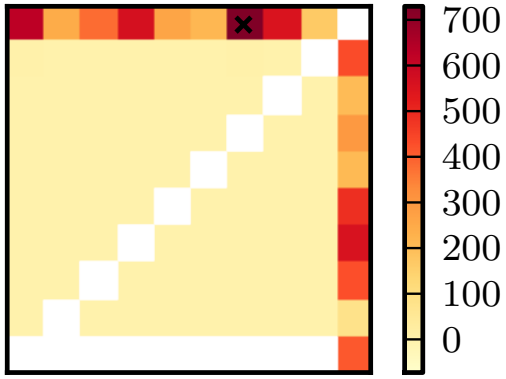
- **Prediction:** entropy of predicted matrix
  $$f(q) = H[R]$$
- **Model:** entropy of posterior over *U* and *V*
  $$f(q) = H[U, V]$$
- **Magnitude search:** mean of found elements
  $$f(q) = R_{ij} + \max_{(k,l) \in \mathcal{P} - (i,j)} \mathbb{E}[R_{kl}]$$
- **Search:** expected number of positives found
  $$f(q) = \mathbb{1}(R_{ij} \in +) + \max_{(k,l) \in \mathcal{P} - (i,j)} \mathbb{P}(R_{kl} \in +)$$

Other work

- Only deals with **Prediction** goal
- Substantial amount of work on active learning for recommender systems, especially the new user case
- Little for general matrix factorization settings:
  - Silva & Carin, KDD 2012
    - assumes fully factorized distribution: more limited model
    - handles much larger datasets
  - Rish & Tesauro, ISAIM 2008 workshop
    - uses max-margin matrix factorization
    - picks points near the boundary

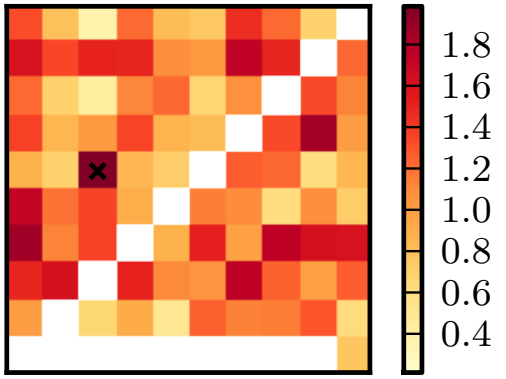
Toy problems

MCMC

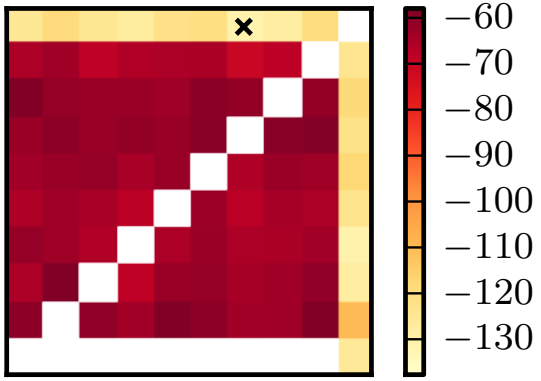


$\mathrm{Var}[R_{ij}]$

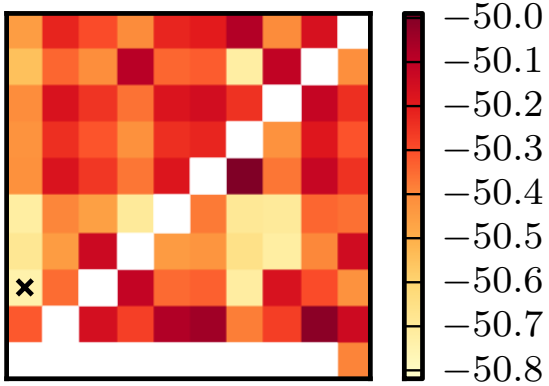
Matrix normal variational



$\mathrm{Var}_q[R_{ij}]$



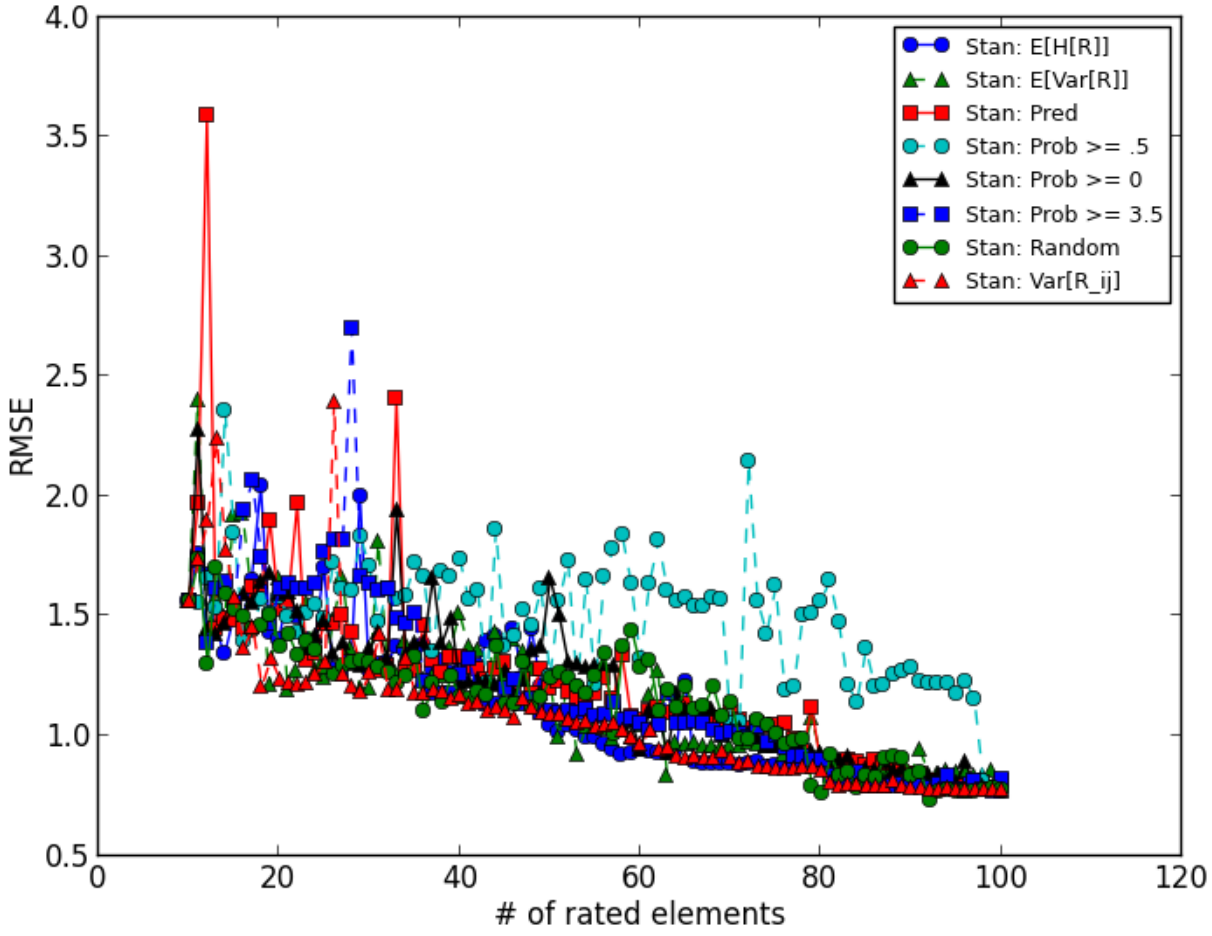
$\mathbb{E}\left[H\left[R\right]\right]$



$\mathbb{E}_q\left[H\left[U,V\right]\right]$

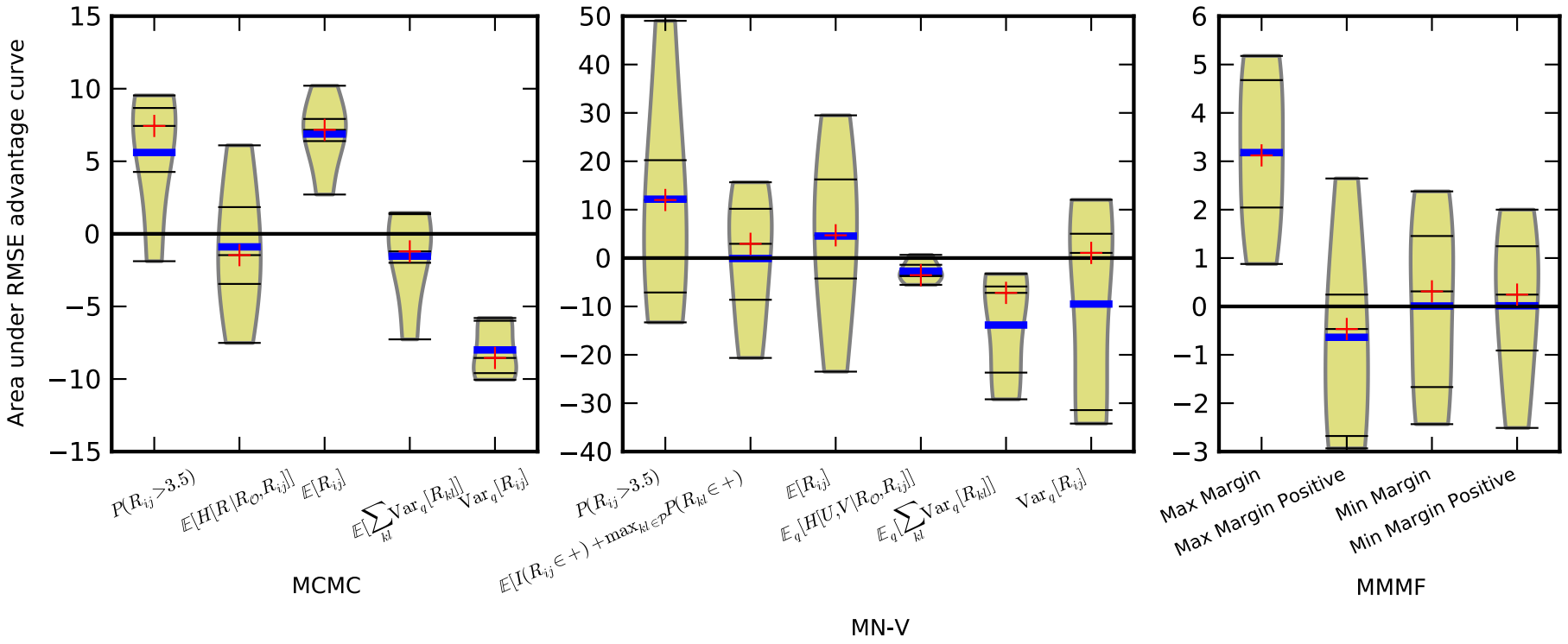
Toy problems

Prediction results on 10x10 rank-4 matrices, vals 1 to 5.



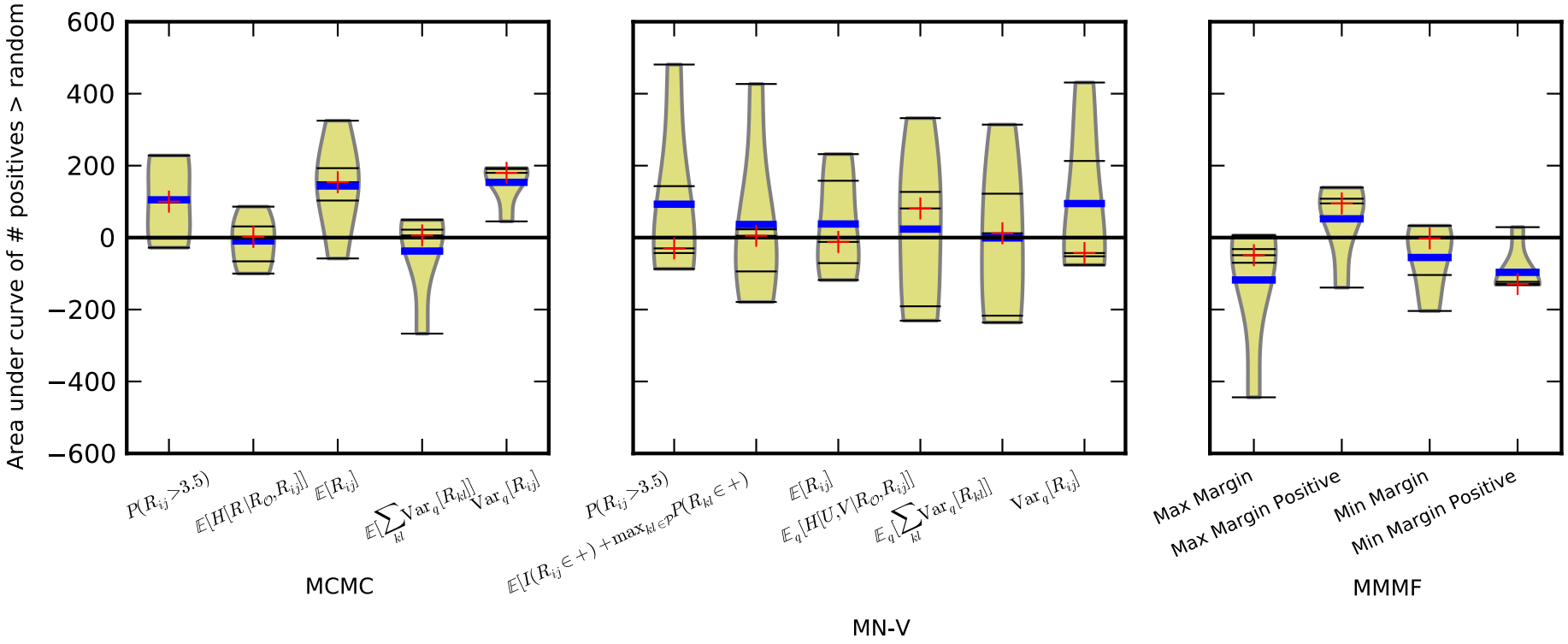
Toy problems

Prediction results on 10x10 rank-4 matrices, vals 1 to 5.



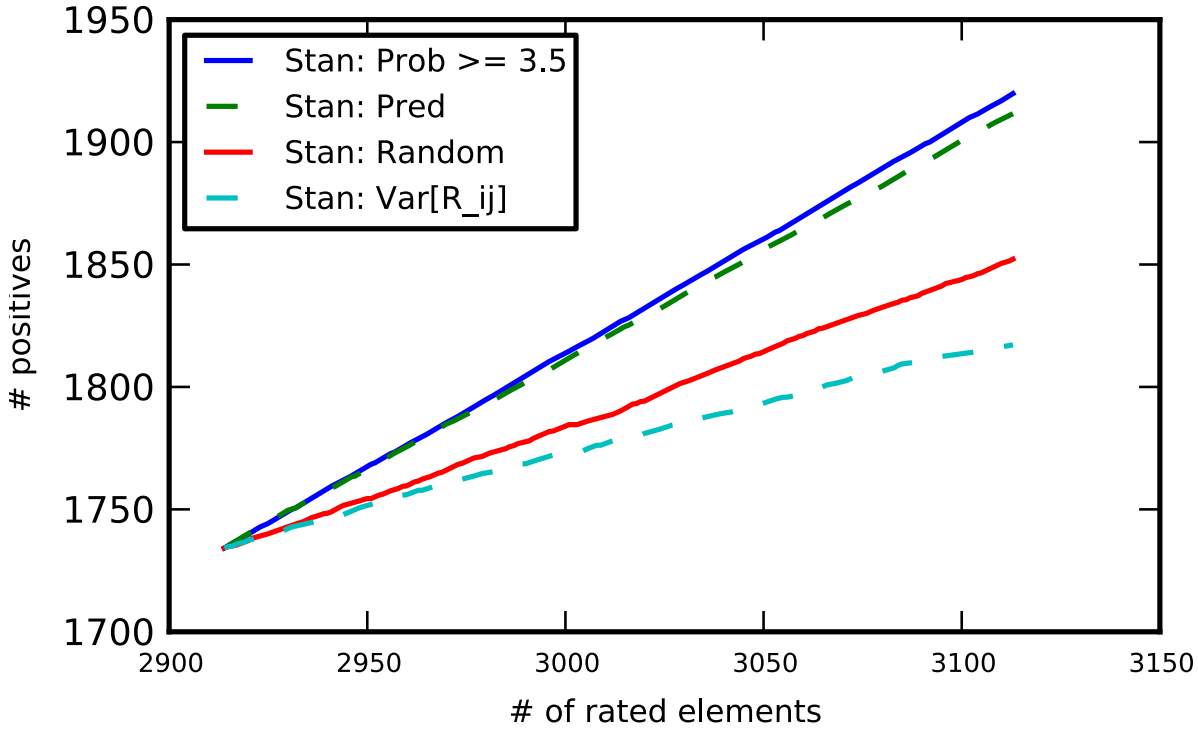
Toy problems

Search results on 10x10 rank-4 matrices, vals 1 to 5.



MovieLens

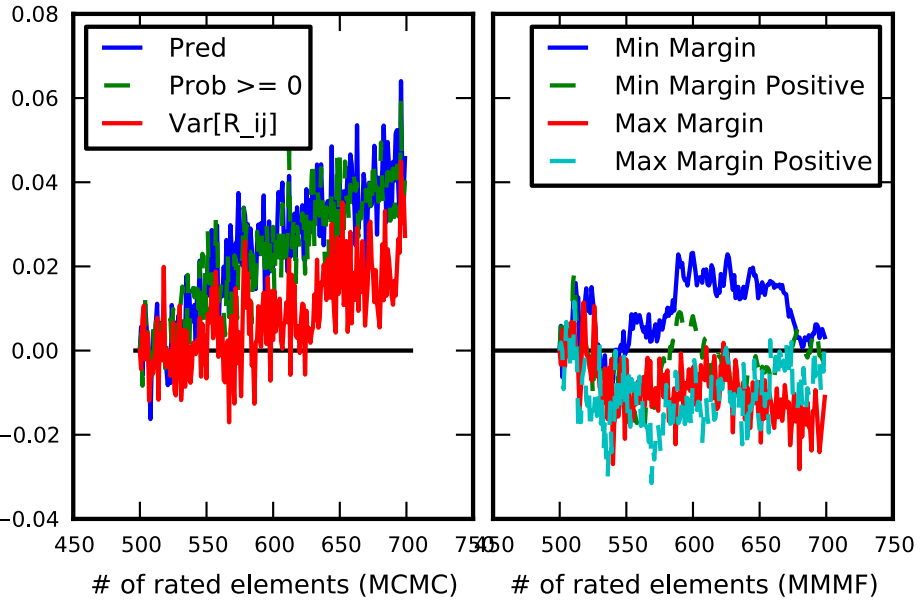
Most of MovieLens-100k: 472 users x 413 movies, ~60k ratings. Start with 5% known; test on a different 5%.

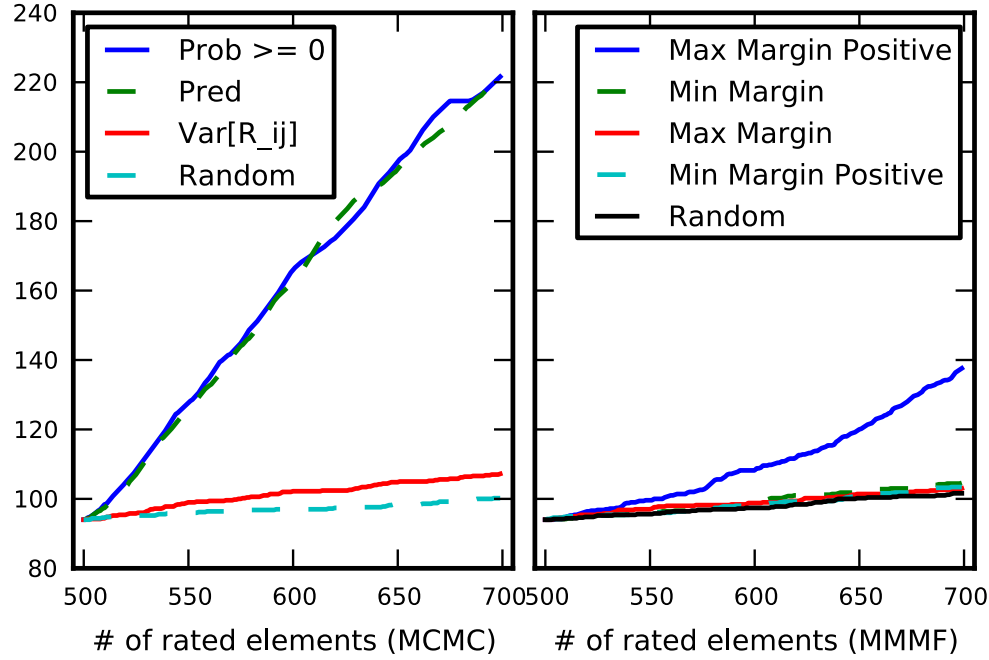


DrugBank

Predict interactions between drugs and “targets.”

- Used a subset of 94 drugs x 425 targets: 4% positive.
- Start with 500 points known: one interaction per drug, enough non-interaction so every column has an entry.
- Test on 500 positives, 1000 negatives; run for 200 steps.





Future work

- Scalability:
  - schemes for choosing points to evaluate in lookahead
  - minibatch/parallel MCMC sampling
- Other, more restricted variational approximations
- Batch selection criteria
- Integrate with side information on points
  - e.g. via GP priors on covariance matrices

Summary

- Collaborative prediction via matrix completion
- Active learning/search to support data collection
- Need distribution information for the criteria:
  - Variational approximations
  - MCMC sampling
- Experiments
  - Toy problems
  - MovieLens
  - DrugBank